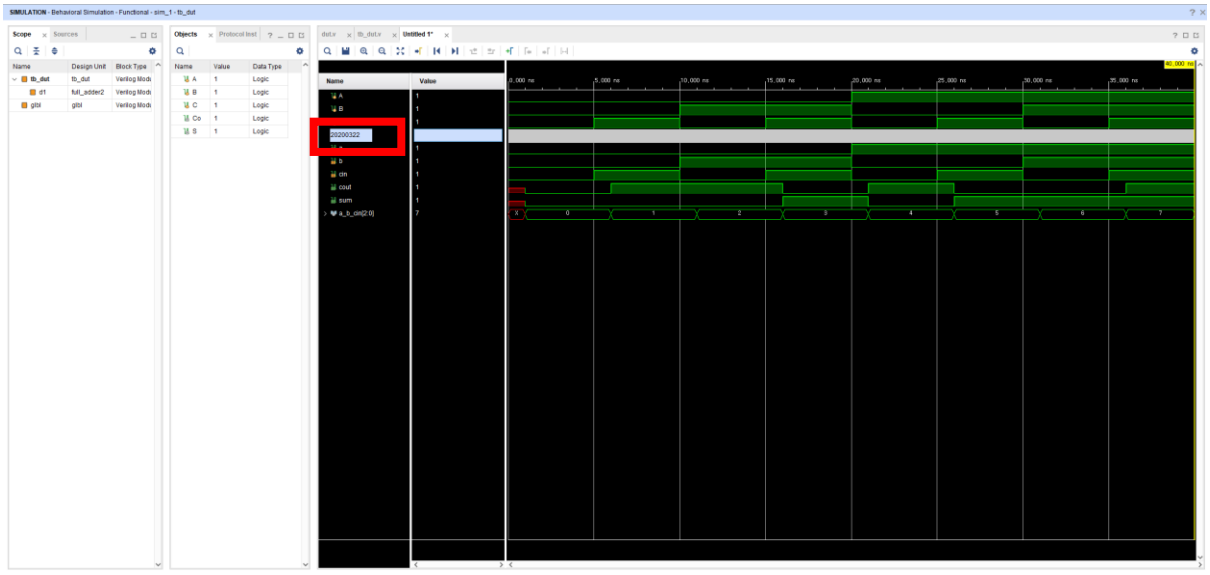
설계과제 HW2

학번 및 이름: **20160394 임효상**

* 문제 3, 4, 5번은 같은 testbench를 사용할 수 있습니다. 1, 2번의 답을 자필로 적어 스캔하여 제출하고, 나머지는 설계한 모듈과 testbench의 verilog 코드를 첨부하고 각 시뮬레이션 결과를 모든 입력 case를 커버할 수 있도록 capture하여 스캔본과 한 파일에 첨부하여 제출하세요. 이때, divider에 학번을 적으세요.



1. Full adder의 진리표를 완성하세요.

A	B	Cin	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

2. Full adder의 Cout과 Sum에 대해 카르노 맵을 그리고 간소화된 논리식을 적으세요.

B,Cin / A	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Cout

B,Cin / A	00	01	11	10
0	0	1	0	1
1	1	0	1	0

SUM

Cout = $BC_{in} + AC_{in} + AB$

Sun = $AB'C_{in}' + A'B'C_{in} + ABC_{in} + A'BC_{in}'$

$= A'(B'C_{in} + BC_{in}') + A(B'C' + BC)$

$= A'(B \text{ xor } C_{in})' + A(B \text{ xor } C)$

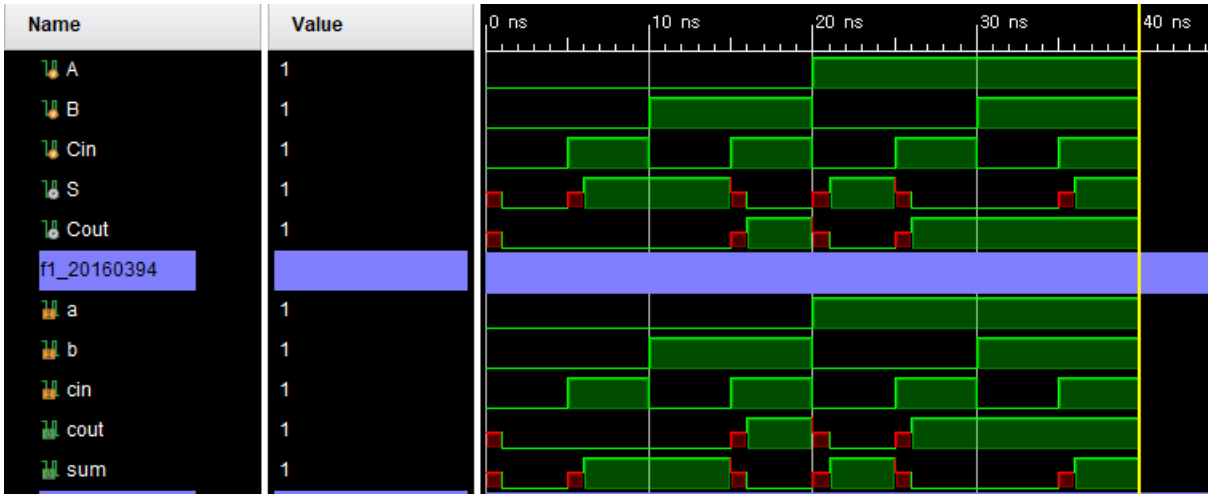
$= A \text{ xor } B \text{ xor } C_{in}$

문제 3,4,5 번에서 사용한 testbench 는 다음과 같습니다.

```
23  module tb_fa();
24
25  reg A, B, Cin;
26  wire S, Cout;
27  fa f1 (.a (A), .b (B), .cin(Cin), .sum(S), .cout(Cout));
28  fa2 f2 (.a (A), .b (B), .cin(Cin), .sum(S), .cout(Cout));
29  fa3 f3 (.a (A), .b (B), .cin(Cin), .sum(S), .cout(Cout));
30
31  initial begin
32  A = 0;
33  B = 0;
34  Cin = 0;
35  #5
36  A = 0;
37  B = 0;
38  Cin = 1;
39  #5
40  A = 0;
41  B = 1;
42  Cin = 0;
43  #5
44  A = 0;
45  B = 1;
46  Cin = 1;
47  #5
48  A = 1;
49  B = 0;
50  Cin = 0;
51  #5
52  A = 1;
53  B = 0;
54  Cin = 1;
55  #5
56  A = 1;
57  B = 1;
58  Cin = 0;
59  #5
60  A = 1;
61  B = 1;
62  Cin = 1;
63  end
64
65  endmodule
66
```

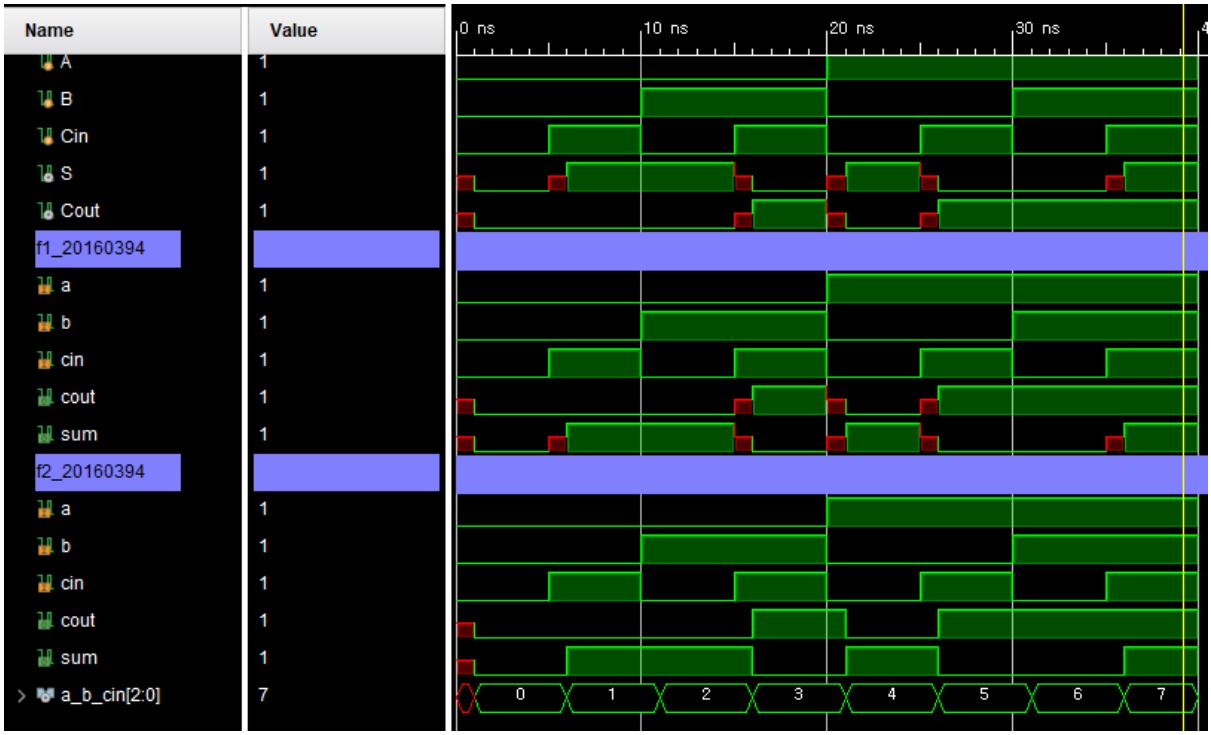
3. Full adder 의 논리식을 continuous assignment 통해 설계하고 모든 입력을 커버하는 testbench 를 작성하여 시뮬레이션하세요. 단, 1 time unit delay 를 줄 것.

```
23  module fa(a, b, cin, cout, sum);
24
25  input a;
26  input b;
27  input cin;
28  output cout, sum;
29
30  assign #1 cout = (b & cin) | (a & cin) | (a & b);
31  assign #1 sum = a ^b ^ cin;
32
33
34  endmodule
```



4. 다음과 같이 A, B, Cin을 concatenate하여 a_b_cin이 정의했을 때 case 문을 이용하여 full adder module을 설계하고 모든 입력을 커버하는 testbench를 작성하여 시뮬레이션하세요.

```
23      module fa2(a, b, cin, cout, sum);
24
25      input a,b,cin;
26      output reg cout, sum;
27      wire[2:0] a_b_cin;
28
29      assign #1 a_b_cin = {a,b,cin};
30
31      always @(a_b_cin)
32      begin
33          case(a_b_cin)
34              3'b000:begin
35              cout = 0;
36              sum = 0;
37              end
38
39              3'b001:begin
40              cout = 0;
41              sum = 1;
42              end
43
44              3'b010:begin
45              cout = 0;
46              sum = 1;
47              end
48
49              3'b011:begin
50              cout = 1;
51              sum = 0;
52              end
53
54              3'b100:begin
55              cout = 0;
56              sum = 1;
57              end
58
59              3'b101:begin
60              cout = 1;
61              sum = 0;
62              end
63
64              3'b110:begin
65              cout = 1;
66              sum = 0;
67              end
68
69              3'b111:begin
70              cout = 1;
71              sum = 1;
72              end
73
74              3'b000:begin
75              cout = 0;
76              sum = 0;
77              end
78          endcase
79      end
80
81      endmodule
```


Name
Value
A
B
Cin
S
Cout
f1_20160394
a
b
cin
cout
sum
f2_20160394
a
b
cin
cout
sum
a_b_cin[2:0]
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
7
0 ns
10 ns
20 ns
30 ns
0
1
2
3
4
5
6
7

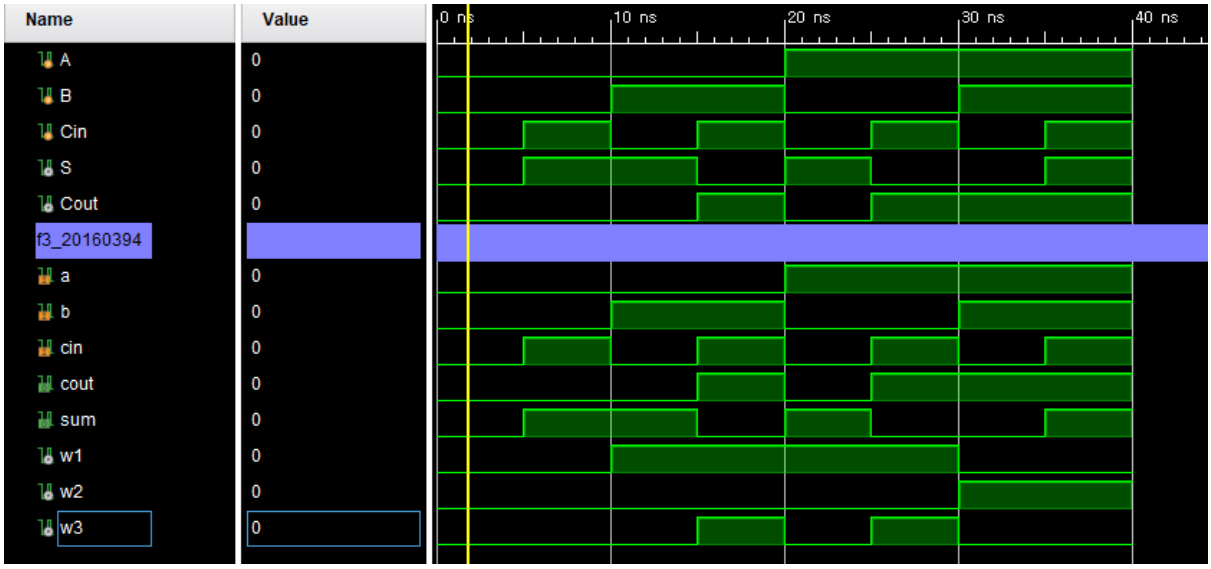
5. Hierarchical design은 submodule을 이용한 설계를 의미합니다. Half adder module을 submodule로 사용하는 full adder를 설계하고 모든 입력을 커버하는 testbench를 작성하여 시뮬레이션하세요. 이때 half adder는 procedural assignment를 이용하여 설계하고 full adder는 continuous assignment를 이용하여 설계하세요.

Full adder

```
23      module fa3(a, b, cin, cout, sum );
24
25      input a,b,cin;
26      output cout, sum;
27
28      wire w1, w2, w3;
29
30      ha h1(.A (a), .B (b), .S(w1), .C(w2));
31      ha h2(.A (w1), .B (cin), .S(sum), .C(w3));
32
33      assign cout = w2 | w3;
34
35      endmodule
```

Half adder

```
22      module ha(A,B,S,C);
23
24      input A,B;
25      output S, C;
26      reg r1,r2;
27
28      assign S = r1;
29      assign C = r2;
30
31      always @(*)
32      begin
33          r1 = A ^ B;
34          r2 = A & B;
35      end
36
37      endmodule
38
```


Name
Value
0 ns
10 ns
20 ns
30 ns
40 ns
A
0
B
0
Cin
0
S
0
Cout
0
f3_20160394
a
0
b
0
cin
0
cout
0
sum
0
w1
0
w2
0
w3
0